UNITED STATES PATENT APPLICATION
FOR

PREBOOT EXECUTION BOOTLOADING

INVENTORS:

REMESH SETH NAIR,
a citizen of Malaysia

ALAN PREVIN TERES ALEXIS,
a citizen of Malaysia

PREPARED BY:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1030
(303) 740-1980

EXPRESS MAIL CERTIFICATE OF MAILING
"Express Mail" Mailing No. EV 331619503 US

I hereby certify that I am causing the above-referenced correspondence to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated below and that this paper or fee has been addressed to Mail Stop Patent Application, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Date of Deposit: December 31, 2003

Name of Person Mailing Correspondence: Krista Mathieson

Krista Mathieson
Signature

December 31, 2003
Date

PREBOOT EXECUTION BOOTLOADING

FIELD

[0001] An embodiment of the invention relates to computer execution in general, and more specifically to preboot execution bootloading.

BACKGROUND

[0002] In computer processing, the actual speed of operation for users is based on numerous factors. One factor that may be of concern is the boot up speed. Boot up performance is a benchmark that is often used to compare operational speed of computer systems.

[0003] While much of boot up performance is a function of how fast the processor of in a system performs operations, an additional variable is the choice of boot up media and the efficiency of the boot up process. A network is one of the media types that may be utilized in booting up operating systems. A typical network boot may rely on the use of PXE (preboot execution environment) processes to enable a preboot environment. This is done to provide a network interface for downloading the OS (operating system) boot image.

[0004] However, conventional methods of network booting of an operating system may be inefficient. A conventional process generally may include setting up a ramdisk, downloading a network bootstrap program (NBP) to the ramdisk from a server, and running network bootstrap program to load the operating system. Thus, conventional remote bootloading may require that certain time-consuming operations be completed before the operating system can be initialized.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] The invention may be best understood by referring to the following description and accompanying drawings that are used to illustrate embodiments of the invention. In the drawings:

[0006] **Figure 1** illustrates an embodiment of a software stack for remote boot up;

[0007] **Figure 2** illustrates an embodiment of operations of a client and server in a remote boot up process;

[0008] **Figure 3** contains flowcharts to illustrate an embodiment of a boot up process;

[0009] **Figure 4** illustrates an embodiment of a computer environment; and

[0010] **Figure 5** illustrates an embodiment of a computer network.

DETAILED DESCRIPTION

[0011] A method and apparatus are described for preboot execution bootloading.

[0012] Before describing an exemplary environment in which various embodiments of the present invention may be implemented, certain terms that will be used in this application will be briefly defined:

[0013] As used herein, "boot" means to load software to start a computer. For example, because the operating system (OS) is used for running all other programs, the OS is loaded during the boot process.

[0014] As used herein, "ramdisk" or "RAM disk" means a virtual computer drive created by allocating or setting aside a portion of the random access memory (RAM) for use as if it were a hard drive or similar drive.

[0015] As used herein, "remote booting" or "network booting" describes a process of booting a computer via a network.

[0016] Under an embodiment of the invention, a computer is remotely booted up by using the preboot environment to load up the operating system. Under an embodiment of the invention, the PXE (preboot execution environment) is utilized to directly load network data into system memory. Under an embodiment of the invention, a PXE API (application programming interface) includes bootloading functionality.

[0017] PXE is a protocol established by Intel Corporation that enables computers to boot through a network. PXE processes may, for example, be stored in the ROM of network cards. When the computer boots up, the BIOS loads the PXE ROM in

the memory and executes it. PXE processes are described in the Preboot Execution Environment Specification 2.1, September 20, 1999, of Intel Corporation.

[0018] PXE allows a workstation to boot from a server on a network prior to booting the operating system on the local hard drive. PXE may be utilized to set up a common set of pre-boot protocol services within a client to ensure that network-based booting is accomplished through an industry standard protocol for communication with the server.

[0019] In a conventional system, a downloaded network bootstrap program (NBP) is required to be presented with a uniform and consistent pre-boot operating environment within the booting client. A conventional system may enable processes such as remote new system setup, remote emergency boot, and network boot processes. Such capabilities may use PXE to form a heterogeneous network, to acquire a network address from a DHCP (dynamic host configuration protocol) server, and to set up a ramdisk within system memory. The ramdisk may then be used to download the network bootstrap program. The network bootstrap program may download the operating system boot, which then may run the bootloading program to load the operating system.

[0020] Under an embodiment of the invention, bootloader functionality for a computer is incorporated into the computer's preboot environment, such as in PXE. Under an embodiment of the invention, PXE is used to load the operating system. The incorporation of such functionality into the environment may be utilized to increase the speed of network booting of an operating system.

[0021] Under an embodiment of the invention, a user may make use of the bootloader functionality of a preboot environment by providing a destination memory

location where the network data may be copied. Under an embodiment of the invention, the PXE may be used to directly load network data into system memory through the use of the UNDI (universal network device interface) network interface.

[0022] Under an embodiment of the invention, a preboot environment is extended to include bootloading, which may allow for standardized implementation of network booting processes. In one embodiment, the PXE API (application programming interface) may be extended to include bootloading functionality. Under an embodiment of the invention, PXE bootloader functionality may allow for efficient remote booting by eliminating the need for a network bootstrapping program.

[0023] **Figure 1** illustrates an embodiment of a software stack for a remote boot up. A conventional software stack **105** provides for PXE ROM runtime **115**. This is followed by the TFTP (trivial file transfer protocol), the UDP (user datagram protocol), and the pre-boot API **120**. The software stack **105** then includes the TFTP/MTFTP (multicast trivial file transfer protocol) **125** for PXE and the DHCP **130** (dynamic host configuration protocol) for IP address allocation. The stack then provides the UDP/IGMP (user datagram protocol/Internet group management protocol) **135**. The stack then includes the IP (Internet protocol) and ARP (address resolution protocol) **140**. This is followed by the base code **145**, the UNDI (universal network device interface) **155**, and the universal network driver **165**. Overlapping with the base code **145**, UNDI **155**, and network driver is the ISR (interrupt service routine) handler **150**.

[0024] An embodiment of a software stack **110** is modified to include a boot load API **170**. Under an embodiment of the invention, the inclusion of the boot load API

enables the client to proceed with a network boot without requiring a network bootstrap program.

[0025] **Figure 2** illustrates an embodiment of operations of a client and server in a remote boot up process. In this illustration, the operations of a PXE client **205** and a PXE server **210** under an embodiment of the invention are shown. In this embodiment, the process includes the PXE client sending a DHCP discovery packet with a PXE boot tag (a network protocol identifier) **215** to the PXE server. The PXE server then sends a DCHP offer packet **220**. After determining that the IP address is legitimate, the PXE client sends a DHCP request packet **225** to the PXE server. The PXE server returns a DCHP acknowledge packet **230**. The PXE client makes a request for a boot memory region **235**. The PXE server reads the boot file and fills memory to the end of the boot file **240**. The PXE server then responds with the boot load data and the memory region **245**. The PXE client than loads the boot load data to the end of the boot file **250**. The client machine thus receives the information and loads the boot image directly into the specified memory region in system memory. This process continues until the end of the boot file. Upon detecting the end of the boot file, the client jumps to the allocated OS memory region to run the operating system **255**.

[0026] **Figure 3** contains flowcharts to illustrate an embodiment of a boot up process. In a conventional PXE remote boot up process, a PXE client uses the DHCP to discover a PXE client tag **305**. After obtaining an IP address, the PXE client then downloads the network bootstrap program **310**. After downloading, the client runs the network bootstrap program to create a ramdisk **315**. The ramdisk is used as the media to

download the OS boot image **320**. The client then boots up the OS boot image **325** and starts the operating system **330**.

[0027] In comparison, under an embodiment of the invention, a client may use the DHCP to discover a PXE boot tag (network protocol identifier) **335**. In this embodiment, the client can then request the memory region address **340** and boot the OS image data to place in the allocated memory region **345**. The client then can jump to the OS execution address **350** and proceed to start the operating system **355**. The operation thus may allow more efficient OS booting and system initialization without the use of a network bootstrap program.

[0028] **Figure 4** illustrates an embodiment of an exemplary computer environment. Under an embodiment of the invention, a computer **400** comprises a bus **405** or other communication means for communicating information, and a processing means such as one or more physical processors **410** (shown as **411** through **412**) coupled with the first bus **405** for processing information. Each of the physical processors may include multiple logical processors, and the logical processors may operate in parallel in the execution of drivers. Each processor may include an execution unit and logic for the operation of certain functions.

[0029] The computer **400** further comprises a random access memory (RAM) or other dynamic storage device as a main memory **415** for storing information and instructions to be executed by the processors **410**. RAM memory may include dynamic random access memory (DRAM) and static dynamic random access memory (SRAM). Main memory **415** also may be used for storing temporary variables or other intermediate information during execution of instructions by the processors **410**. The computer **400**

also may comprise a read only memory (ROM) **420** and/or other static storage device for storing static information and instructions for the processor **410**.

[0030] A data storage device **425** may also be coupled to the bus **405** of the computer **400** for storing information and instructions. The data storage device **425** may include a magnetic disk or optical disc and its corresponding drive, flash memory or other nonvolatile memory, or other memory device. Such elements may be combined together or may be separate components, and utilize parts of other elements of the computer **400**.

[0031] The computer **400** may also be coupled via the bus **405** to a display device **430**, such as a liquid crystal display (LCD) or other display technology, for displaying information to an end user. In some environments, the display device may be a touch-screen that is also utilized as at least a part of an input device. In some environments, display device **430** may be or may include an auditory device, such as a speaker to produce voice or other auditory information. An input device **440** may be coupled to the bus **405** for communicating information and/or command selections to the processor **410**. In various implementations, input device **440** may be a keyboard, a keypad, a touch-screen and stylus, a voice-activated system, or other input device, or combinations of such devices. Another type of user input device that may be included is a cursor control device **445**, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor **410** and for controlling cursor movement on display device **430**.

[0032] A communication device **450** may also be coupled to the bus **405**. Depending upon the particular implementation, the communication device **450** may include a transceiver, a wireless modem, a network interface card, or other interface

device. The computer **400** may be linked to a network or to other devices using the communication device **450**, which may include links to the Internet, a local area network, or another environment. Under an embodiment of the invention, the computer **400** may be linked to a server, including a PXE server, via the communication device **450**.

[0033] **Figure 5** illustrates an embodiment of a computer network environment **500**. According to one embodiment, a network **505** is present. The network **505** may include a physical network, a logical network, a wide area network (WAN), a local area network (LAN), the Internet, an intranet, or any other computer network or computer communication environment. A computer system or physical machine **510** (machine) may include a single machine, or multiple machines coupled with each other directly, or via a network or device, or via other physical and/or logical links. The machine **510** may include a PXE client.

[0034] According to one embodiment, the machine **510** may include a modem **515**, a network interface **520**, and any other type of connection that allows communication with the network **505**. A communication via the network **505** may include the use of any of wired or wireless carrier and protocols, including radio frequency (RF), satellite, microwave, Bluetooth, optical, infrared, cable, laser, or other process. The machine **510** may communicate with various remote machines via the network **505**. The remote machines are shown as, for example, remote machine 1 **525** and remote machine **530**. The possible remote machines may include a server **535**. The server **535** may include a PXE server. Under an embodiment of the invention, the machine **510** may boot an operating system remotely in conjunction with server **535**.

[0035] In the description above, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without some of these specific details. In other instances, well-known structures and devices are shown in block diagram form.

[0036] The present invention may include various processes. The processes of the present invention may be performed by hardware components or may be embodied in machine-executable instructions, which may be used to cause a general-purpose or special-purpose processor or logic circuits programmed with the instructions to perform the processes. Alternatively, the processes may be performed by a combination of hardware and software.

[0037] Portions of the present invention may be provided as a computer program product, which may include a machine-readable medium having stored thereon instructions, which may be used to program a computer (or other electronic devices) to perform a process according to the present invention. The machine-readable medium may include, but is not limited to, floppy diskettes, optical disks, CD-ROMs, and magneto-optical disks, ROMs (read-only memory), RAMs (random access memory), EPROMs (erasable programmable read-only memory), EEPROMs (electrically-erasable programmable read-only memory, magnet or optical cards, flash memory, or other type of media/machine-readable medium suitable for storing electronic instructions. Moreover, the present invention may also be downloaded as a computer program product, wherein the program may be transferred from a remote computer to a requesting

computer by way of data signals embodied in a carrier wave or other propagation medium via a communication link (e.g., a modem or network connection).

[0038] Many of the methods are described in their most basic form, but processes can be added to or deleted from any of the methods and information can be added or subtracted from any of the described messages without departing from the basic scope of the present invention. It will be apparent to those skilled in the art that many further modifications and adaptations can be made. The particular embodiments are not provided to limit the invention but to illustrate it. The scope of the present invention is not to be determined by the specific examples provided above but only by the claims below.

[0039] It should also be appreciated that reference throughout this specification to "one embodiment" or "an embodiment" means that a particular feature may be included in the practice of the invention. Similarly, it should be appreciated that in the foregoing description of exemplary embodiments of the invention, various features of the invention are sometimes grouped together in a single embodiment, figure, or description thereof for the purpose of streamlining the disclosure and aiding in the understanding of one or more of the various inventive aspects. This method of disclosure, however, is not to be interpreted as reflecting an intention that the claimed invention requires more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed embodiment. Thus, the claims are hereby expressly incorporated into this description, with each claim standing on its own as a separate embodiment of this invention.